01-06-06

AF

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/880,631/Conf #5913
Filing Date	June 12, 2001
First Named Inventor	Wenting Tang
Art Unit	2157
Examiner Name	G. G. Todd
Total Number of Pages in This Submission: 55	Attorney Docket Number: 10010812-1

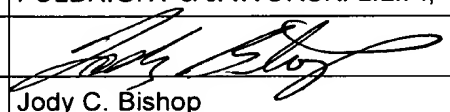
ENCLOSURES (*Check all that apply*)

- [ ] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment/Reply
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Reply to Missing Parts/ Incomplete Application
  - [ ] Reply to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [x] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) (please Identify below):

2 Return Postcards
Certificate of Mail

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	FULBRIGHT & JAWORSKI L.L.P., Attorney, Agent for Applicant		
Signature			
Printed name	Jody C. Bishop		
Date	January 5, 2006	Reg. No.	44,034

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 10010812-1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Wenting Tang et al.

Application No.: 09/880,631

Confirmation No.: 5913

Filed: June 12, 2001

Art Unit: 2157

For: METHOD AND SYSTEM FOR MODULAR TRANSMISSION CONTROL PROTOCOL (TCP) RARE-HANDOFF DESIGN IN A STREAMS BASED TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL (TCP/IP) IMPLEMENTATION

Examiner: G. G. Todd

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Notification of Non-Compliant Appeal Brief mailed December 21, 2005, Appellant hereby submits this Appeal Brief that corrects the identified deficiencies of the Appeal Brief filed September 19, 2005. Specifically, this Appeal Brief identifies a related appeal in section II hereof and includes Appendices B and C. Otherwise, this Appeal Brief is identical to the Appeal Brief filed September 19, 2005. Thus, this Appeal Brief is in furtherance of the Notice of Appeal filed on July 19, 2005 and is believed to be in full compliance with 37 C.F.R. § 41.37.

The fees required under § 41.20(b)(2) were dealt with in the Appeal Brief filed September 19, 2005. No further fees are believed to be due for this modified brief in response to the Notification of Non-Compliant Appeal Brief of December 21, 2005.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

Appellant did not identify any related appeals in the Appeal Brief filed September 19, 2005. Because the present application does not reference co-pending U.S. Patent Application Serial No. 09/880,632, Appellant did not consider such co-pending application as related. However, because the Examiner asserts in the Notification of Non-Compliant Appeal Brief of December 21, 2005 that the Appeal Brief should identify the appeal of co-pending U.S. Patent Application Serial No. 09/880,632 as a related appeal, Appellant hereby does so.

A notice of appeal was filed for co-pending U.S. Patent Application Serial No. 09/880,632 (hereafter "the '632 application") on July 18, 2005. The claims of the '632 application are rejected on the same grounds as the claims of the present application, i.e., as being anticipated by U.S. Patent No. 6,775,692 issued to Albert et al ("*Albert*"). Thus, the appeal of the '632 application (and particularly the Board's interpretation of *Albert*) may affect, be affected by, or have a bearing on the Board's decision in this appeal.

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 37 claims pending in application.

B. Current Status of Claims

1. Claims canceled: None
2. Claims withdrawn from consideration but not canceled: None
3. Claims pending: 1-37
4. Claims allowed: None
5. Claims rejected: 1-37

C. Claims On Appeal

The claims on appeal are claims 1-37

IV. STATUS OF AMENDMENTS

A first Office Action was mailed for this application September 22, 2004. In response, Applicant filed an Amendment on December 22, 2004, which presented an amendment to claim 5. A Final Office Action was then mailed April 20, 2005. Applicant did not file an amendment in response to the Final Office Action, but instead filed a Notice of Appeal, which this brief supports. Thus, the claims on appeal are those claims rejected in the Final Office Action, and a listing of those claims are provided in Appendix A hereto.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

According to one claimed embodiment of the present invention, a method of TCP state migration in a communication network comprises establishing a TCP/IP communication session between a client computer (e.g., client 410 of Figure 4) and a first server computer (e.g., server 450 of Figure 4). The first server computer is part of a plurality of server computers forming a web cluster containing information (e.g., web cluster 490 of Figure 4). The communication session is established for the transfer of data contained within the information. The method further comprises handing off the communication session to a selected server computer (e.g., server 452 of Figure 4, and see page 23, lines 9-13 of the specification) from the first server computer over a persistent control channel using TCP handoff modules (e.g., Upper TCP module 522 and Bottom TCP module 524 of Figure 5C, and *see* page 8, line 25 – page 11, line 6, and page 29, line 27 – page 30, line 6 of the specification) that are dynamically loadable (*see* page 17, line 24 – page 18, line 15 of the specification) within TCP/IP stacks in operating systems located at both the first server computer and the selected server computer, that implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocol (*see* page 23, line 21 – page 26, line 29 of the specification). The method further comprises migrating a first TCP state of the first server computer to the selected server computer, and a second TCP state of the selected server computer to the first server computer over the control channel (e.g., page 10, line 26 – page 11, line 28 of the specification).

In one embodiment, establishing the TCP/IP communication session further comprises receiving a SYN packet from the client at a first BTCP module located at the first server computer (block 1010 of Figure 10); sending the SYN packet upstream to a first TCP module located above the first BTCP module in a first operating system of the first server

computer (block 1020 of Figure 10); receiving a first SYN/ACK packet from the first TCP module (block 1030 of Figure 10); parsing the first initial TCP state from the first SYN/ACK packet, including a first initial sequence number for the first TCP module associated with the TCP/IP communication session (block 1040 of Figure 10); sending the SYN/ACK packet to the client (block 1050 of Figure 10); receiving an ACK packet from the client at the first BTCP module (block 1060 of Figure 10); sending the ACK packet to the first TCP module (block 1070 of Figure 10); receiving a web request packet associated with the TCP/IP communication session at the first BTCP module at the first server computer (block 1080 of Figure 10); and storing the SYN, ACK and the web request packet at the first server computer (block 1090 of Figure 10).

In one embodiment, handing off the communication session further comprises examining content of the web request packet; determining which of the plurality of server computers, a selected server computer, can best process the WEB request packet, based on the content (page 10, lines 7-16 of the specification); sending a handoff request from said first BTCP module to a second BTCP module at the selected server computer over the control channel, if the selected server computer is not the first server computer; including the SYN packet and the ACK packet in the handoff request packet; changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module; sending said SYN packet to said second TCP module; receiving a second SYN/ACK packet at said second BTCP module; parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session; changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module; updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session; sending said ACK packet that is updated to said second TCP module; and sending a handoff acknowledgment message to said first BTCP module.

According to another claimed embodiment, a method of TCP state migration in a communication network comprises establishing a TCP/IP communication session between a client computer (e.g., client 410 of Figure 4) and a first server computer (e.g., server 450 of Figure 4). The first server computer is part of a plurality of server computers forming a web

cluster containing information (e.g., web cluster 490 of Figure 4), and the communication session is established for the transfer of data contained within the information. The method further comprises monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of the first server computer associated with the TCP/IP communication session, at a first bottom-TCP (BTCP) module at the first server computer (Bottom TCP module 524 of Figure 5 and BTCP module 830 of Figure 8, and *see* page 9, lines 16-20 and page 11, lines 8-11 of the specification). The method further comprises receiving a web request associated with the TCP/IP communication session at the first BTCP module at the first server computer (block 910 of Figure 9 and block 1310 of Figure 13, and *see* page 10, lines 7-8 of the specification). The method further comprises examining content of the web request, and determining which of the plurality of server computers ("a selected server computer") can best process the web request, based on the content (block 930 of Figure 9, and *see* page 10, lines 13-16 of the specification). The method further comprises handing off the communication session to the selected server (e.g., server 452 of Figure 4) computer from the first server computer over a persistent control channel, if the selected server computer is not the first server computer (*see* page 10, line 26 – page 11, line 28 and page 23, lines 9-13 of the specification). The method further comprises monitoring traffic associated with handing off the TCP/IP communication session to understand a second initial TCP state of the selected server computer associated with the TCP/IP communication session, at a second BTCP module at the selected server computer (e.g., BTCP module 870 of Figure 8, and *see* page 12, lines 1-13 of the specification). The method further comprises migrating the first initial TCP state to the selected server computer over the control channel, such that the second BTCP module can calculate a first TCP state for the first server computer in the TCP/IP communication session (e.g., page 12, line 15 – page 13, line 8 of the specification). The method further comprises sending a second initial TCP state of the selected server computer to the first BTCP module (e.g., BTCP module 830 of Figure 8), such that the first BTCP module can calculate a second TCP state for the selected server computer in the TCP/IP communication session. The method further comprises forwarding data packets received at the first BTCP module from the client to the selected server computer, by changing the data packets to reflect the second TCP state and a second IP address of the selected server computer (e.g., block 1320 of Figure 13, and *see* page 12, line 15 – page 13, line 8 of the specification). The method further comprises sending response packets from the selected server computer directly to the client computer

(*see* Figures 3 and 4) by changing the response packets to reflect the first TCP state and a first IP address of the first server computer (e.g., block 1440 of Figure 14, and *see* page 12, line 15- page 13, line 8 of the specification). And, the method further comprises terminating the TCP/IP communication session at the first server computer when the TCP/IP communication session is closed (e.g., page 13, lines 10-19).

According to another claimed embodiment, a server computer comprises an upper TCP (UTCP) module (e.g., UTCP module 522 of Figure 5C, and UTCP modules 810 and 850 of Figure 8) located above a TCP module (e.g., TCP module 520 of Figures 5B and 5C, and TCP modules 820 and 860 in Figure 8) in an operating system of the server computer. The server computer further comprises a bottom TCP (BTCP) module (e.g., BTCP module 524 of Figure 5C, and BTCP modules 830 and 870 of Figure 8) located below the TCP module. The UTCP, TCP, and BTCP modules implement a method of handing off a communication session between a first node (e.g., server 450 of Figure 4, and "front-end" node of Figure 8) and second node (e.g., server 452 of Figure 4, and "back-end" node of Figure 8) in a cluster network (e.g., cluster 490 of Figure 4) that works within the kernel level of an existing TCP/IP protocol, by migrating TCP states associated with the first and second nodes (*see* page 10, line 26 – page 12, line 13 of the specification).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-37 are rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,775,692 issued to Albert et al (hereinafter "*Albert*").

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board reverse the outstanding rejections in light of the remarks contained herein. Below, Appellant argues many of the rejected claims separately. Thus, Appellant respectfully asserts that separately argued claims do not stand or fall together, *see* 37 C.F.R. § 41.37(c)(1)(vii).

To anticipate a claim under 35 U.S.C. § 102, a single reference must teach every element of the claim, *see* M.P.E.P. § 2131. Appellant respectfully submits that *Albert* fails to teach each and every element of claims 1-37, as discussed further below.

Independent Claim 1

Independent claim 1 recites:

> In a communication network, a method of TCP state migration comprising the steps of:
>
> a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;
>
> b) handing off said communication session to a selected server computer from said first server computer over a persistent control channel using TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, that implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocol; and
>
> c) migrating a first TCP state of said first server computer to said selected server computer, and a second TCP state of said selected server computer to said first server computer over said control channel. (Emphasis added).

Albert fails to teach all elements of independent claim 1. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 1. Accordingly, without conceding that *Albert* teaches any of the other elements of claim 1, *Albert* fails to teach at least those elements described further below. Specifically, *Albert* fails to teach at least:

A) TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer;

B) handing off said communication session; and

C) a persistent control channel.

A. Albert fails to teach TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer as recited by claim 1

Albert fails to teach TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems, as recited in claim 1. In general, *Albert* is directed to a

"system and method for proxying a connection using a distributed architecture". Col. 5, lines 1-3. *Albert* provides at col. 5, lines 3-15:

> A service manager attracts from forwarding agents packets that are sent by clients attempting to set up connections that are to be proxied. For each proxied connection, the service manager establishes a connection with an appropriate server and transfers data between the server and the client. At some point, the service manager may determine that the connection has reached a state in which it is appropriate to no longer proxy the connection and to allow packets to flow directly between the client and the server. At that point, the service manager sends instructions to the forwarding agents for adjusting packet sequence numbers so that packets may be forwarded between the client and the server.

Albert teaches a system that includes clients, forwarding agents, service managers, and servers. A forwarding agent receives communication from a client, and interacts with a service manager to determine which server to forward the communication. The forwarding agent then forwards the communication to the server dictated by the service manager. The service managers may instruct forwarding agents, via wildcard affinities, as to which flows the service managers are interested in. Upon forwarding agents receiving a flow matching a wildcard affinity, the forwarding agent notifies the service manager, which in turn instructs the forwarding agent how to handle the flow (e.g., where to forward the packets, etc.). In this way, the service manager can make decisions and instruct the forwarding agents in order to perform load-balancing among a plurality of servers. The forwarding agents act as intermediaries in which all communication between a client and a server (which may be determined by the service manager) flows through the forwarding agents. *See* col. 7, line 20 – col. 14, line 64 of *Albert*; and *see* Figs. 2A, 3A-3C, 4-5, and 11-12 of *Albert*.

The Final Office Action cites col. 14, line 65 – col. 15, line 27 (SYN/ACK packets) of *Albert* in support of its assertion that *Albert* teaches the above-identified element of claim 1, *see* page 3 of the Final Office Action. Col. 14, line 65 – col. 15, line 27 of *Albert* provides:

> FIG. 5 is a diagram illustrating how a service manager provides instructions to two separate forwarding agents for handling a connection. A client 500 sends a SYN packet to a first forwarding agent 502. Forwarding agent 502 has previously received a wildcard affinity from a service manager 504 on a dedicated connection on which service manager 504 multicasts wildcard affinities to forwarding agents. As a result of the wildcard match, forwarding agent 502 encapsulates the SYN packet and forwards it to service

> manager 504. Service manager 504 receives the SYN packet and returns it to forwarding agent 502 along with a fixed affinity specifying an action to be performed on the packet. The action defined in this example is translating the destination IP address of the packet from a virtual IP address to the IP address of a host 506. Hosts 506 and 507 together implement a virtual machine 510.
>
> Host 1 receives the SYN packet from forwarding agent 1 and returns a SYN ACK packet back to client 500. However, for some reason, the SYN ACK packet from host 1 is routed not through forwarding agent 502, but instead through forwarding agent 512. Forwarding agent 512 receives the SYN ACK and notes that it matches a wildcard affinity corresponding to the flow of packets from host 506 to client 500. Forwarding agent 512 encapsulates the SYN ACK packet and sends it to service manager 504. Service manager 504 defines an action for the SYN ACK packet and includes that action in a second fixed affinity which it sends along with the encapsulated SYN ACK packet back to forwarding agent 512. Forwarding agent 512 then sends the SYN ACK packet on to client 500 where it is processed.

The above portion of *Albert* teaches that upon receiving a SYN packet, the forwarding agent 502 forwards such SYN packet to a service manager if a predefined wildcard affinity matches the SYN packet. The service manager returns the SYN packet to the forwarding agent with a fixed affinity specifying an action to be performed on the packet. Host 1 receives the SYN packet from the forwarding agent 502 and returns a SYN ACK packet through forwarding agent 512 to client 500. Again, because the SYN ACK packet matches a predefined wildcard affinity, forwarding agent 512 sends the SYN ACK packet to the service manager. The service manager returns the SYN ACK packet to the forwarding agent 512 with a fixed affinity, and the forwarding agent 512 sends the SYN ACK packet on to client 500.

The above portion of *Albert* in no way teaches "TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer". The Final Office Action appears to contend that the forwarding agent 502 of *Albert* is the recited first server computer of claim 1, and the Host 1 of *Albert* is the recited selected server computer of claim 1. *Albert* fails, however, to teach that either of the forwarding agent 502 or Host 1 use a TCP handoff module that is dynamically loadable within a TCP/IP stack in its operating system, and certainly fails to teach that both of the elements comprise such TCP handoff modules that are dynamically loadable within TCP/IP stacks in their operating systems.

Appellant presented the above argument in the Amendment dated December 22, 2004. In response to this argument, the Final Office Action further asserts at page 14 thereof:

> A module can be defined as "A self-contained functional unit which is used with a larger system. A software module is a part of a program that performs a particular task. A hardware module can be a packaged unit that attaches to a system." Albert is teaching a forwarding agent (first server computer) having an interface (at least Fig. 2C) and connected to a plurality of back-end nodes or servers (at least Fig. 2A; selected servers) through which communication with a client is allowed, thus a front-end module or program performing a particular task directing client requests to servers and TCP flow and connection control and handing off, transparent to the client, the TCP connection from the forwarding agent to the back-end node or server (at least col. 13, lines 10-29; col. 8, lines 17-39; col. 12, lines 22-35).

The above-portion of the Final Office Action appears to quote a definition for "module," but fails to identify the source of the definition. Further, the above-quoted portion of the Final Office Action appears to assert that the forwarding agent of *Albert* corresponds to the recited "first server computer" in claim 1, and the back-end nodes or servers of *Albert* correspond to the "selected server" recited in claim 1. It remains unclear in the Final Office Action, however, what in *Albert* corresponds to the recited "TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer". Again, *Albert* does not teach the recited dynamically loadable TCP handoff modules at the forwarding agent (first server computer) and back-end nodes (selected server). While *Albert* teaches a forwarding agent that is capable of receiving requests and forwarding them to back-end nodes, it does not teach that the forwarding agent and back-end nodes comprise TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems. *Albert* simply provides no teaching regarding any dynamically loadable modules.

B. Albert fails to teach handing off said communication session

Further, *Albert* does not teach "handing off said communication session", as recited by claim 1. Instead, the forwarding agent acts as an intermediary to forward communication that it receives to an appropriate back-end server, but the communication session is not "handed off" to the back-end server. That is, the forwarding agent forwards packets to the

appropriate back-end server, but it does not hand off an established communication session to the back-end server.

C. Albert fails to teach a persistent control channel as recited by claim 1

Further, it is unclear what teaching of *Albert* is believed by the Final Office Action to teach a "persistent control channel", as recited by claim 1. Appellant respectfully submits that *Albert* does not teach a persistent control channel as recited by claim 1. Rather, in *Albert,* communication for any given flow can be received by any number of different forwarding agents, which interact with the service managers to forward the communication to the appropriate back-end server. *Albert* does not teach handing off a communication session from the forwarding agent to the back-end server over a persistent control channel.

Accordingly, *Albert* fails to teach at least the above-identified elements of claim 1, and therefore claim 1 is not anticipated under 35 U.S.C. § 102 by *Albert.* As such, Appellant respectfully requests that the rejection of claim 1 be overturned.

Independent Claim 11

Independent claim 11 recites:

> In a communication network, a method of TCP state migration comprising the steps of:
> a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;
> b) monitoring traffic associated with establishing said TCP/IP communication session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer;
> c) receiving a web request associated with said TCP/IP communication session at said first BTCP module at said first server computer;
> d) examining content of said web request;
> e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;
> f) handing off said communication session to said selected server computer from said first server computer over a persistent control channel, if

said selected server computer is not said first server computer;

g) monitoring traffic associated with handing off said TCP/IP communication session to understand a second initial TCP state of said selected server computer associated with said TCP/IP communication session, at a second BTCP module at said selected server computer;

h) migrating said first initial TCP state to said selected server computer over said control channel, such that said second BTCP module can calculate a first TCP state for said first server computer in said TCP/IP communication session;

i) sending a second initial TCP state of said selected server computer to said first BTCP module, such that said first BTCP module can calculate a second TCP state for said selected server computer in said TCP/IP communication session;

j) forwarding data packets received at said first BTCP module from said client to said selected server computer, by changing said data packets to reflect said second TCP state and a second IP address of said selected server computer;

k) sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer; and

l) terminating said TCP/IP communication session at said first server computer when said TCP/IP communication session is closed.

Albert fails to teach all elements of independent claim 11. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 11. Accordingly, without conceding that *Albert* teaches any of the other elements of claim 11, *Albert* fails to teach at least those elements described further below. Specifically, *Albert* fails to teach at least:

A) a first bottom-TCP (BTCP) module at said first server computer, and a second BTCP module at said selected server computer;

B) examining content of said web request and determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;

C) sending response packets from said selected server computer directly to said client computer;

D) handing off said communication session; and

E) a persistent control channel.

A. Albert fails to teach a first bottom-TCP (BTCP) module at said first server computer and a second BTCP module at the selected server computer

Albert fails to teach a first bottom TCP (BTCP) module at the first server computer. *Albert* also fails to teach a second bottom TCP (BTCP) module at the selected server computer. Indeed, *Albert* does not teach a modularized solution, and thus fails to teach any modules whatsoever.

In response to the above argument, the Final Office Action further asserts at page 14:

> Albert is teaching a forwarding agent having an interface (at least Fig. 2C) and connected to a plurality of back-end nodes or servers (at least Fig. 2A) through which communication with a client is allowed, thus a front-end module or program performing a particular task directing client requests to servers and offering SYN ACK packets (BTCP) and TCP flow (at least col. 13, lines 10-29; col. 8, lines 17-39)

It is unclear from the above-quoted portion of the Final Office Action what in *Albert* the Examiner contends corresponds to the recited first bottom TCP (BTCP) module at the first server computer and the recited second BTCP module at the selected server computer. As described above with claim 1, the Final Office Action appears to contend that the forwarding agent of *Albert* corresponds to the recited first server computer, and the back-end servers of *Albert* correspond to the recited selected server computer. However, it is unclear what teaching of *Albert* the Final Office Action believes to teach BTCP modules of the forwarding agent and back-end servers. The above-quoted portion of the Final Office Action appears to assert that the SYN ACK packets provide the recited BTCP module (because "BTCP" appears in parenthesis following this language). However, this is nonsensical. The SYN ACK packets in *Albert* are not modules, but are instead merely packets that are communicated.

Albert simply provides no teaching of a first bottom-TCP (BTCP) module at a first server computer or a second BTCP module at the selected server computer, as recited by claim 11.

B. Albert fails to teach examining content of said web request and determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content, as recited by claim 11

Additionally, independent claim 11 further recites "e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content" (emphasis added). *Albert* fails to teach determining a web server that can best process a received web request based on the content of such web request. Rather, as described further below, *Albert* teaches using pre-defined wildcard affinities to select a server to handle a request based on information included in a SYN packet, such as the requesting client's IP address, rather than selecting a server based on the content of a web request.

The Final Office Action cites col. 9, lines 10-34 and 45-58 of *Albert* in support of its assertion that *Albert* teaches this element of claim 11, *see* page 11 of the Office Action. Col. 9, lines 10-34 and 45-58 of *Albert* provides:

> In addition to specifying instructions for each flow, service managers must also obtain information about each new flow from the forwarding agents. For example, when a service manager provides load balancing through a set of forwarding agents, the service manager uses fixed affinities to provide specific instructions to the forwarding agents detailing where packets for each load balanced flow are to be forwarded. In addition to providing those specific instructions, the service manager also provides general instructions to each forwarding agent that specify which new flows the service manager is interested in seeing. These general instructions are provided using wildcard affinities. Wildcard affinities, which are described in detail below, specify sets of flows that are of interest to a service manager. In one embodiment, this is done by specifying subnet masks that determine sets of source and destination IP addresses that will be forwarded to a service manager. In addition, ports or sets of ports and protocol may be specified in wildcard affinity as well. As is described further below, the use of wildcard affinities enables separate service managers to be configured to provide services for different sets of flows. Each service manager specifies the flows of interest to it and other service managers handle other flows. In this manner, service managers can be configured in parallel to share load.
>
> * * *
>
> In the case of load balancing, service managers send wildcard affinities to forwarding agents. The wildcard affinities specify destination IP addresses that correspond to virtual IP addresses of server clusters that are to be load balanced by the service manager. The forwarding agents then forward new packets sent to those virtual IP addresses to the appropriate service manager. The service manager selects a server from the server cluster and then the

> service manager sends a fixed affinity to each forwarding agent that instructs the forwarding agent to forward packets for that specific flow to the selected server in the cluster. Forwarding agents may also forward packets for purposes other than load balancing. Packets may be forwarded to real IP addresses as well as virtual IP addresses.

Albert does not teach determining a web server that can best process a received web request based on the content of such web request. Rather, *Albert* teaches pre-setting a wildcard affinity based on a client's IP address. For instance, the above portion of *Albert* teaches that wildcard affinities specify sets of flows that are of interest to a service manager. The wildcard affinities are pre-selected (e.g., before even receiving a request from a client), to specify a subnet mask that identifies a set of source and destination IP addresses. Thus, for instance, a particular IP address corresponding to a client of interest can be identified by a wildcard affinity.

As described further in *Albert* at col. 12, line 6 – col. 14, line 15, a Syn packet is used to identify whether the flow matches a wildcard affinity, in which case it is forwarded to the service manager for determination of how to handle the flow. Thus, the service manager in *Albert* selects a server responsive to receipt of a Syn packet, which is an initial connection establishment packet sent before it is even known what the request will be (i.e., before knowing that it is a web request). That is, the Syn packet upon which *Albert* selects a server, does not include content of a web request. Rather, the Syn packet includes source and destination IP addresses, from which it is determined by the forwarding agents whether such packet corresponds to a pre-set wildcard affinity. Thus, the back-end server is not selected in *Albert* based on the content of such web request, as the server is selected based on other information (e.g., source IP address) included in the Syn packet.

C. Albert fails to teach sending response packets from said selected server computer directly to said client computer, as recited by claim 11

Claim 11 recites "k) sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer" (emphasis added). *Albert* fails to teach a system in which the selected server responds directly to the client computer, but rather all communication flows through the forwarding agents. That is, the forwarding agents

and service manager of *Albert* act as a front-end node through which all communication from clients is received in order to determine the back-end server (or host) to which the communication should be sent, and all return communication from the back-end server (or host) is sent back through the forwarding agents to the client. Thus, this configuration is similar to that of FIGURE 1 described in the present application, wherein the forwarding agents/service manager of *Albert* provide a load balancer through which all communication between the clients and the back-end servers (hosts) flows. *Albert* does not teach a configuration in which a selected back-end server sends a response directly to the client computer.

D. Albert fails to teach handing off said communication session

Further, *Albert* does not teach "handing off said communication session", as recited by claim 11. Instead, the forwarding agent acts as an intermediary to forward communication that it receives to an appropriate back-end server, but the communication session is not "handed off" to the back-end server. That is, the forwarding agent forwards packets to the appropriate back-end server, but it does not hand off an established communication session to the back-end server.

E. Albert fails to teach a persistent control channel as recited by claim 11

Albert does not teach a persistent control channel as recited by claim 11. Rather, in *Albert,* communication for any given flow can be received by any number of different forwarding agents, which interact with the service managers to forward the communication to the appropriate back-end server. *Albert* does not teach handing off a communication session from the forwarding agent to the back-end server over a persistent control channel.

Accordingly, *Albert* fails to teach at least the above-identified elements of claim 11, and therefore claim 11 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 11 be overturned.

Independent Claim 26

Independent claim 26 recites:

> A server computer comprising:
> an upper TCP (UTCP) module located above a TCP module in an operating system of said server computer;
> a bottom TCP (BTCP) module located below said TCP module, said UTCP, TCP, and BTCP modules implementing a method of handing off a communication session between a first node and second node in a cluster network that works within the kernel level of an existing TCP/IP protocol, by migrating TCP states associated with said first and second nodes.

Albert fails to teach all elements of independent claim 26. As described above, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 26. Accordingly, without conceding that *Albert* teaches any of the other elements of claim 26, *Albert* fails to teach at least those elements described further below.

For example, independent claim 26 recites "an upper TCP (UTCP) module located above a TCP module in an operating system of said server computer" (emphasis added). *Albert* fails to teach such an upper TCP (UTCP) module located above a TCP module in an operating system of a server computer.

Further, claim 26 recites "a bottom TCP (BTCP) module located below said TCP module" (emphasis added). *Albert* also fails to teach such a bottom TCP (BTCP) module located below the TCP module in an operating system.

Indeed, *Albert* does not teach modules at all in the operating systems of its computers. While *Albert* describes forwarding agents, service manager, and host computers, it fails to teach that any of those include a module located above or below a TCP module in an operating system.

Further still, claim 26 recites "said UTCP, TCP, and BTCP modules implementing a method of handing off a communication session between a first node and second node in a cluster network that works within the kernel level of an existing TCP/IP protocol" (emphasis added). *Albert* does not teach this further element of claim 26. That is, *Albert* does not teach handing off a communication session, as recited by claim 26.

Accordingly, *Albert* fails to teach at least the above-identified element of claim 26, and therefore claim 26 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 26 be overturned.

Dependent Claim 2

Claim 2 depends from claim 1 and thus inherits all elements of claim 1. Accordingly, claim 2 is allowable over *Albert* at least for the reasons discussed above with claim 1. Additionally, claim 2 further recites:

> The method as described in Claim 1, wherein said step a) comprises the steps of:
> receiving a SYN packet from said client at a first BTCP module located at said first server computer;
> sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer;
> receiving a first SYN/ACK packet from said first TCP module;
> parsing said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with said TCP/IP communication session;
> sending said SYN/ACK packet to said client;
> receiving an ACK packet from said client at said first BTCP module;
> sending said ACK packet to said first TCP module;
> receiving a web request packet associated with said TCP/IP communication session at said first BTCP module at said first server computer;
> storing said SYN, ACK and said web request packet at said first server computer.

Albert fails to teach all of the further elements of claim 2. For instance, *Albert* fails to teach at least the recited first BTCP module located at said first server computer. Accordingly, claim 2 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 2 be overturned.

Dependent Claim 3

Claim 3 depends from claim 2, which depends from claim 1, and thus claim 3 inherits all elements of claims 1 and 2. Accordingly, claim 3 is allowable over *Albert* at least for the reasons discussed above with claims 1 and 2. Additionally, claim 3 further recites:

The method as described in Claim 2, wherein said step b) comprises the steps of:

examining content of said web request packet;

determining which of said plurality of server computers, a selected server computer, can best process said WEB request packet, based on said content;

sending a handoff request from said first BTCP module to a second BTCP module at said selected server computer over said control channel, if said selected server computer is not said first server computer;

including said SYN packet and said ACK packet in said handoff request packet;

changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module;

sending said SYN packet to said second TCP module;

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session;

changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module;

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;

sending said ACK packet that is updated to said second TCP module; and

sending a handoff acknowledgment message to said first BTCP module.

Albert fails to teach all of the further elements of claim 3. For instance, *Albert* fails to teach at least the recited second BTCP module at said selected server computer. Further, as discussed below with independent claim 11, *Albert* fails to teach examining content of said web request packet and determining which of said plurality of server computers, a selected server computer, can best process said WEB request packet, based on said content. Accordingly, claim 3 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 3 be overturned.

Dependent Claim 4

Claim 4 depends from claim 3, which depends from claim 2 which depends from 1, and thus claim 4 inherits all elements of claims 1, 2, and 3. Accordingly, claim 4 is allowable over *Albert* at least for the reasons discussed above with claims 1-3. Additionally, claim 4 further recites:

> The method as described in Claim 3, wherein step c) comprises the steps of:
>
> monitoring traffic associated with establishing said TCP/IP communication session in step a), at said first BTCP module, to parse a first initial TCP state of said first server computer, said first initial TCP state associated with said TCP/IP communication session; and
>
> migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request packet, said first initial TCP state including a first sequence number, such that said second BTCP module can calculate said first TCP state for said first server computer in said TCP/IP communication session.

Albert fails to teach all of the further elements of claim 4. For instance, *Albert* fails to teach at least the recited "migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request packet, said first initial TCP state including a first sequence number". Accordingly, claim 4 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 4 be overturned.

Dependent Claim 5

Claim 5 depends from claim 3, which depends from claim 2 which depends from 1, and thus claim 5 inherits all elements of claims 1, 2, and 3. Accordingly, claim 5 is allowable over *Albert* at least for the reasons discussed above with claims 1-3. Additionally, claim 5 further recites:

> The method as described in Claim 3, wherein step c) comprises the steps of:
>
> monitoring traffic associated with handing off said TCP/IP communication session at said second BTCP module, to parse a second initial TCP state of said selected server computer, said second initial TCP state associated with said TCP/IP communication session; and
>
> migrating said second initial TCP state of said selected server computer to said first BTCP module by including said second initial TCP state in said handoff acknowledgment packet, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said selected server computer in said TCP/IP communication session.

Albert fails to teach all of the further elements of claim 5. For instance, *Albert* fails to teach at least the recited "migrating said second initial TCP state of said selected server computer to said first BTCP module by including said second initial TCP state in said

handoff acknowledgment packet, said second initial TCP state including a second initial sequence number". Accordingly, claim 5 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 5 be overturned.

Dependent Claim 6

Claim 6 depends from claim 2, which depends from claim 1, and thus claim 6 inherits all elements of claims 1 and 2. Accordingly, claim 6 is allowable over *Albert* at least for the reasons discussed above with claims 1 and 2. Additionally, claim 6 further recites:

> The method as described in Claim 2, comprising the further steps of:
> intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session; and
> holding said connection indication message at said first UTCP module.

Albert fails to teach all of the further elements of claim 6. For instance, *Albert* fails to teach at least the recited first upper TCP UTCP) module. Accordingly, claim 6 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 6 be overturned.

Dependent Claim 7

Claim 7 depends from claim 6, which depends from claim 2 which depends from claim 1, and thus claim 7 inherits all elements of claims 1, 2, and 6. Accordingly, claim 7 is allowable over *Albert* at least for the reasons discussed above with claims 1, 2, and 6. Additionally, claim 7 further recites:

> The method as described in Claim 6, wherein said method comprises the further steps of:
> sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module;
> discarding said connection indication message at said first UTCP module;
> receiving incoming data packets from said client at said first BTCP module;
> changing said destination addresses of said incoming data packets to said second IP address;
> updating sequence numbers and TCP checksum in said data packets to

reflect said second TCP state of said selected server computer; and
forwarding said data packets to said selected server computer.

Albert fails to teach all of the further elements of claim 7. For instance, *Albert* fails to teach at least the recited "sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module". Accordingly, claim 7 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 7 be overturned.

Dependent Claim 8

Claim 8 depends from claim 6, which depends from claim 2 which depends from claim 1, and thus claim 8 inherits all elements of claims 1, 2, and 6. Accordingly, claim 8 is allowable over *Albert* at least for the reasons discussed above with claims 1, 2, and 6. Additionally, claim 8 further recites:

> The method as described in Claim 6, comprising the further steps of:
> sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer;
> sending incoming data packets, including said web request packet, from said client, received at said first BTCP module, upstream.

Albert fails to teach all of the further elements of claim 8. For instance, *Albert* fails to teach at least the recited "sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer". Accordingly, claim 8 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 8 be overturned.

Dependent Claim 9

Claim 9 depends from claim 1 and thus inherits all elements of claim 1. Accordingly, claim 9 is allowable over *Albert* at least for the reasons discussed above with claim 1. Additionally, claim 9 further recites:

> The method as described in Claim 1, comprising the further step of:
> intercepting outgoing response packets from said selected server

computer at a second bottom TCP (BTCP) module located at said selected server computer;

changing source addresses of said response packets to a first IP address of said first server computer;

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said response packets to said client.

Albert fails to teach all of the further elements of claim 9. For instance, *Albert* fails to teach at least the recited second bottom TCP (BTCP) module located at said selected server computer. Accordingly, claim 9 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 9 be overturned.

Dependent Claim 10

Claim 10 depends from claim 1 and thus inherits all elements of claim 1. Accordingly, claim 10 is allowable over *Albert* at least for the reasons discussed above with claim 1. Additionally, claim 10 further recites:

The method as described in Claim 1, comprising the further steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module;

understanding when said communication session is closed at said second server computer;

sending a termination message to said first server computer over said control channel;

terminating said TCP/IP communication session at said first server computer by terminating a forwarding mode at said first BTCP module; and

freeing data resources associated with said communication session at said first server computer.

Albert fails to teach all of the further elements of claim 10. For instance, *Albert* fails to teach at least the recited first and second bottom TCP (BTCP) modules. Accordingly, claim 10 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 10 be overturned.

Dependent Claim 12

Claim 12 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 12 is allowable over *Albert* at least for the reasons discussed above with claim 11. Additionally, claim 12 further recites:

The method as described in Claim 11, wherein said step a) comprises the steps of:

receiving a packet from said client at said first BTCP module;

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer;

receiving a first SYN/ACK packet from said first TCP module;

parsing said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with, said TCP/IP communication session;

sending said SYN/ACK packet to said client;

receiving an ACK packet from said client at said first BTCP module;

sending said ACK packet to said first TCP module;

storing said SYN, ACK and said web request at said first server computer.

Albert fails to teach all of the further elements of claim 12. For instance, *Albert* fails to teach at least the recited "sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer". Accordingly, claim 12 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 12 be overturned.

Dependent Claim 13

Claim 13 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 13 is allowable over *Albert* at least for the reasons discussed above with claim 11. Additionally, claim 13 further recites:

The method as described in Claim 11, wherein said step e) comprises the steps of:

sending a handoff request packet from said first BTCP module to said second BTCP module over said control channel;

including said SYN packet and said ACK packet in said handoff request packet;

changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module;

sending said SYN packet to said second TCP module;

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session;

changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module;

updating said ACK packet to reflect said second TCP state of said

selected server computer in said communication session;
sending said ACK packet that is updated to said second TCP module; and
sending a handoff acknowledgment message to said first BTCP module.

Albert fails to teach all of the further elements of claim 13. For instance, *Albert* fails to teach at least the recited "changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module; ...and sending a handoff acknowledgment message to said first BTCP module" (emphasis added). Accordingly, claim 13 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 13 be overturned.

Dependent Claims 14-15

Claims 14-15 each depend from claim 13, which depends from claim 11. Thus, claims 14 and 15 each inherit all elements of claims 11 and 13, and are therefore allowable over *Albert* at least for the reasons discussed above with claims 11 and 13. Therefore, Appellant respectfully requests that the rejection of claims 14 and 15 be overturned.

Dependent Claim 16

Claim 16 depends from claim 13, which depends from claim 11, and thus claim 16 inherits all elements of claims 11 and 13. Accordingly, claim 16 is allowable over *Albert* at least for the reasons discussed above with claims 11 and 13. Additionally, claim 16 further recites:

> The method as described in Claim 13, comprising the further steps of:
> intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session; and
> holding said connection indication message at said first UTCP module.

Albert fails to teach all of the further elements of claim 16. For instance, *Albert* fails to teach at least the recited first upper TCP (UTCP) module. Accordingly, claim 16 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 16 be overturned.

Dependent Claim 17

Claim 17 depends from claim 16, which depends from claim 13 which depends from claim 11, and thus claim 17 inherits all elements of claims 11, 13, and 16. Accordingly, claim 17 is allowable over *Albert* at least for the reasons discussed above with claims 11, 13, and 16. Additionally, claim 17 further recites:

> The method as described in Claim 16, wherein step h) comprises the further steps of:
> sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module;
> discarding said connection indication message at said first UTCP module;
> receiving incoming data packets from said client at said first BTCP module;
> changing said destination addresses of said incoming data packets to said second IP address;
> updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and
> forwarding said data packets to said selected server computer.

Albert fails to teach all of the further elements of claim 17. For instance, *Albert* fails to teach at least the recited "sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module". Accordingly, claim 17 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 17 be overturned.

Dependent Claim 18

Claim 18 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 18 is allowable over *Albert* at least for the reasons discussed above with claim 11. Additionally, claim 18 further recites:

> The method as described in Claim 11, wherein step k) comprises the steps of:
> intercepting outgoing response packets from said selected server computer at said second BTCP module;
> changing source addresses of said response packets to said first IP address;
> updating sequence numbers and TCP checksum in said response

packets to reflect said first TCP state of said first server computer; and
sending said updated response packets to said client.

Albert fails to teach all of the further elements of claim 18. For instance, *Albert* fails to teach at least the recited "intercepting outgoing response packets from said selected server computer at said second BTCP module". Accordingly, claim 18 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 18 be overturned.

Dependent Claim 19

Claim 19 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 19 is allowable over *Albert* at least for the reasons discussed above with claim 11. Additionally, claim 19 further recites:

The method as described in Claim 11, wherein step 1) comprises the steps of:
monitoring TCP/IP control traffic for said communication session at said second BTCP module;
understanding when said communication session is closed at said second server computer;
sending a termination message to said first server computer over said control channel;
terminating a forwarding mode at said first BTCP module; and
freeing data resources associated with said communication session at said first server computer.

Albert fails to teach all of the further elements of claim 19. For instance, *Albert* fails to teach at least the recited "monitoring TCP/IP control traffic for said communication session at said second BTCP module" and "terminating a forwarding mode at said first BTCP module". Accordingly, claim 19 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 19 be overturned.

Dependent Claim 20

Claim 20 depends from claim 16, which depends from claim 13 which depends from claim 11, and thus claim 20 inherits all elements of claims 11, 13, and 16. Accordingly, claim 20 is allowable over *Albert* at least for the reasons discussed above with claims 11, 13, and 16. Additionally, claim 20 further recites:

> The method as described in Claim 16, comprising the further steps of:
> sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer; and
> sending incoming data packets, including said web request, from said client, received at said first BTCP module, upstream.

Albert fails to teach all of the further elements of claim 20. For instance, *Albert* fails to teach at least the recited "sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer". Accordingly, claim 20 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 20 be overturned.

Dependent Claim 21

Claim 21 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 21 is allowable over *Albert* at least for the reasons discussed above with claim 11. Additionally, claim 21 further recites:

> The method as described in Claim 11, wherein each of said plurality of server computers is constructed similarly including BTCP modules located downstream from TCP modules, and UTCP modules located upstream from TCP modules.

Albert fails to teach all of the further elements of claim 21. For instance, *Albert* fails to teach each of its server computers include BTCP modules located downstream from TCP modules, and UTCP modules located upstream from TCP modules. Again, *Albert* fails to teach the recited BTCP and UTCP modules. Accordingly, claim 21 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 21 be overturned.

Dependent Claim 22

Claim 22 depends from claim 12, which depends from claim 11. Thus, claim 22 inherits all elements of claims 11 and 12, and are therefore allowable over *Albert* at least for the reasons discussed above with claims 11 and 12. Therefore, Appellant respectfully requests that the rejection of claim 22 be overturned.

Dependent Claim 23

Claim 23 depends from claim 22, which depends from claim 12 which depends from claim 11. Thus, claim 23 inherits all elements of claims 11, 12, and 22. Accordingly, claim 23 is allowable over *Albert* at least for the reasons discussed above with claims 11, 12, and 22. Additionally, claim 23 further recites:

> The method as described in Claim 22, wherein said control channel allows for communication between all UTCP modules.

Albert fails to teach this further element of claim 23. For instance, *Albert* fails to teach each UTCP modules, and thus fails to teach a control channel that allows communication between all of such UTCP modules. Accordingly, claim 23 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 23 be overturned.

Dependent Claims 24-25

Claims 24-25 each depend from claim 11, and thus claims 24 and 25 each inherit all elements of claim 11. Therefore, claims 24-25 are each allowable over *Albert* at least for the reasons discussed above with claim 11. As such, Appellant respectfully requests that the rejection of claims 24 and 25 be overturned.

Dependent Claim 27

Claim 27 depends from claim 26 and thus inherits all elements of claim 26. Accordingly, claim 27 is allowable over *Albert* at least for the reasons discussed above with claim 26. Additionally, claim 27 further recites:

> The server computer as described in Claim 26, wherein said method comprises the steps of:
> a) establishing a TCP/IP communication session between a client computer and said server computer, said first node, said server computer part of a plurality of server computers forming said cluster network containing information, said communication session established for the transfer of data contained within said information;
> b) receiving a web request associated with said TCP/IP communication session at a first BTCP module at said server computer;

c) examining content of said web request;
d) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;
e) handing off said communication session to said selected server computer from said server computer over a persistent control channel, if said selected server computer is not said server computer; and
f) migrating a first TCP state of said server computer to said selected server computer, and sending a second TCP state of said selected server computer to said server computer over said control channel. (Emphasis added).

Albert fails to teach all of the further elements of claim 27. For instance, as discussed above with claim 11, *Albert* fails to teach determining which of the plurality of server computers, a selected server computer, can best process a web request, based on content of the web request. As also discussed above with claim 11, *Albert* fails to teach handing of a communication session over a persistent control channel. Accordingly, claim 27 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 27 be overturned.

Dependent Claim 28

Claim 28 depends from claim 27, which depends from claim 26, and thus claim 28 inherits all elements of claims 26 and 27. Accordingly, claim 28 is allowable over *Albert* at least for the reasons discussed above with claims 26 and 27. Additionally, claim 28 further recites:

The server computer as described in Claim 27, wherein step a) of said method comprises the steps of:
receiving a SYN packet from said client at said BTCP module;
sending said SYN packet upstream to said TCP module;
receiving a first SYN/ACK packet from said TCP module;
parsing a first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said TCP module associated with said TCP/IP communication session;
sending said SYN/ACK packet to said client;
receiving an ACK packet from said client at said BTCP module;
sending said ACK packet to said TCP module;
storing said SYN, ACK at said server computer.

Albert fails to teach all of the further elements of claim 28. For instance, as discussed above, *Albert* fails to teach the BTCP module, and thus for at least this reason *Albert* fails to

teach the above steps that involve such BTCP module. Accordingly, claim 28 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 28 be overturned.

Dependent Claim 29

Claim 29 depends from claim 28, which depends from claim 27 which depends from claim 26, and thus claim 29 inherits all elements of claims 26-28. Accordingly, claim 29 is allowable over *Albert* at least for the reasons discussed above with claims 26-28. Additionally, claim 29 further recites:

> The server computer as described in Claim 28, wherein said method comprises the steps of:
> sending a handoff request packet from said BTCP module to a second BTCP module over said control channel, said second BTCP module located below a second TCP module in a second operating system at said selected server computer;
> including said SYN packet and said ACK packet in said handoff request;
> receiving a handoff acknowledgment message at said BTCP module from said second BTCP module.

Albert fails to teach all of the further elements of claim 29. For instance, as discussed above, *Albert* fails to teach the recited BTCP modules, and thus for at least this reason *Albert* fails to teach the above steps that involve such BTCP modules. Accordingly, claim 29 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 29 be overturned.

Dependent Claim 30

Claim 30 depends from claim 29, which depends from claim 28 which depends from claim 27 which depends from claim 26, and thus claim 30 inherits all elements of claims 26-29. Accordingly, claim 30 is allowable over *Albert* at least for the reasons discussed above with claims 26-29. Additionally, claim 30 further recites:

> The server computer as described in Claim 29, wherein said step f) of said method comprises the steps of:
> monitoring traffic associated with establishing said TCP/IP communication session in step a), at said BTCP module, to parse a first initial

TCP state of said server computer, said first initial TCP state associated with said TCP/IP communication session; and

migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request, said first initial TCP state including a first sequence number, such that said second BTCP module can calculate said first TCP state for said server computer in said TCP/IP communication session.

Albert fails to teach all of the further elements of claim 30. For instance, as discussed above, *Albert* fails to teach the recited BTCP module and control channel, and thus for at least these reasons *Albert* fails to teach the above steps that involve such BTCP modules and control channel. Accordingly, claim 30 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 30 be overturned.

Dependent Claim 31

Claim 31 depends from claim 29, which depends from claim 28 which depends from claim 27 which depends from claim 26, and thus claim 31 inherits all elements of claims 26-29. Accordingly, claim 31 is allowable over *Albert* at least for the reasons discussed above with claims 26-29. Additionally, claim 31 further recites:

The server computer as described in Claim 29, wherein said method comprises the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session; and

holding said connection indication message at said first UTCP module.

Albert fails to teach all of the further elements of claim 31. For instance, *Albert* fails to teach the recited first upper TCP (UTCP) module, and thus for at least this reason *Albert* fails to teach the above steps that involve such UTCP module. Accordingly, claim 31 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 31 be overturned.

Dependent Claim 32

Claim 32 depends from claim 31, which depends from claim 29 which depends from claim 28 which depends from claim 27 which depends from claim 26, and thus claim 32

inherits all elements of claims 26-29 and 31. Accordingly, claim 32 is allowable over *Albert* at least for the reasons discussed above with claims 26-29 and 31. Additionally, claim 32 further recites:

> The computer system as described in Claim 31, wherein said method comprises the further steps of:
> sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module;
> discarding said connection indication message at said first UTCP module;
> receiving incoming data packets from said client at said first BTCP module;
> changing said destination addresses of said incoming data packets to said second IP address;
> updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and
> forwarding said data packets to said selected server computer.

Albert fails to teach all of the further elements of claim 32. For instance, *Albert* fails to teach the recited "sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module" and "discarding said connection indication message at said first UTCP module". Accordingly, claim 32 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 32 be overturned.

Dependent Claim 33

Claim 33 depends from claim 31, which depends from claim 29 which depends from claim 28 which depends from claim 27 which depends from claim 26, and thus claim 33 inherits all elements of claims 26-29 and 31. Accordingly, claim 33 is allowable over *Albert* at least for the reasons discussed above with claims 26-29 and 31. Additionally, claim 33 further recites:

> The server computer as described in Claim 31, said method comprising the further steps of:
> sending notification from said BTCP module to said UTCP module to release said connection indication message, if said selected server computer is said server computer;
> sending incoming data packets, including said web request, from said client, received at said first BTCP module, upstream.

Albert fails to teach all of the further elements of claim 33. For instance, *Albert* fails to teach the recited BTCP and UTCP modules, and thus for at least this reason *Albert* fails to teach the above steps involving such modules. Accordingly, claim 33 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 33 be overturned.

Dependent Claim 34

Claim 34 depends from claim 26, and thus inherits all elements of claim 26. Accordingly, claim 34 is allowable over *Albert* at least for the reasons discussed above with claim 26. Additionally, claim 34 further recites:

> The server computer as described in Claim 26, said method comprising the further steps of:
> receiving a handoff request from a first BTCP module located at a first server computer within said cluster network over a persistent control channel, said first server computer having established a communication session with a client computer, said communication session established for the transfer of data contained within said server computer, said handoff request including a SYN packet and an ACK packet, said SYN and ACK packet used for establishing said communication session between said client and said first server computer, said ACK packet including a first initial TCP state of said first server computer in said communication session, including a first initial TCP sequence number;
> changing a first destination IP address of said SYN packet to a second IP address of said server computer, at said BTCP module;
> sending said SYN packet to said TCP module;
> receiving a SYN/ACK packet at said second BTCP module;
> parsing a second initial TCP state from second SYN/ACK packet, including a second initial sequence number, for said TCP module, said second initial TCP state associated with a second TCP state for said server computer in said TCP/IP communication session;
> changing a second destination IP address of said ACK packet to said second IP address, at said BTCP module;
> updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;
> sending said ACK packet that is updated to said TCP module; and
> sending a handoff acknowledgment message to said first BTCP module over said control channel.

Albert fails to teach all of the further elements of claim 34. For instance, *Albert* fails to teach at least the recited "receiving a handoff request from a first BTCP module located at a first server computer within said cluster network over a persistent control channel". As

discussed above with claim 11, *Albert* fails to teach a first BTCP module or a persistent control channel. Further, *Albert* fails to teach a "handoff request". Rather, *Albert* merely teaches that its forwarding agents forward packets to a selected back-end server, rather than handing off the TCP communication session. Thus, for at least these reasons, claim 34 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 34 be overturned.

Dependent Claim 35

Claim 35 depends from claim 34, which depends from claim 26, and thus claim 35 inherits all elements of claims 26 and 34. Accordingly, claim 35 is allowable over *Albert* at least for the reasons discussed above with claims 26 and 34. Additionally, claim 35 further recites:

> The server computer as described in Claim 34, wherein said method comprises the further steps of:
>
> monitoring traffic associated with handing off said TCP/IP communication session to said server computer, at said BTCP module, to parse said second initial TCP state of said server computer, said second initial TCP state associated with said TCP/IP communication session; and
>
> sending said second initial TCP state of said server computer to said first BTCP module by including said second initial TCP state in said handoff acknowledgment, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said server computer in said TCP/IP communication session.

Albert fails to teach all of the further elements of claim 35. For instance, *Albert* fails to teach at least the recited BTCP module, thus for at least this reason *Albert* fails to teach the above steps that involve such BTCP module. Accordingly, claim 35 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 35 be overturned.

Dependent Claim 36

Claim 36 depends from claim 34, which depends from claim 26, and thus claim 36 inherits all elements of claims 26 and 34. Accordingly, claim 36 is allowable over *Albert* at least for the reasons discussed above with claims 26 and 34. Additionally, claim 36 further recites:

The server computer as described in Claim 34, wherein said method comprises the further steps of:

intercepting outgoing response packets from said server computer at said second BTCP module;

changing source addresses of said response packets to said first IP address;

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said response packets to said client.

Albert fails to teach all of the further elements of claim 36. For instance, *Albert* fails to teach at least the recited second BTCP module, thus for at least this reason *Albert* fails to teach the above intercepting step that involves such second BTCP module. Accordingly, claim 36 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 36 be overturned.

Dependent Claim 37

Claim 37 depends from claim 34, which depends from claim 26, and thus claim 37 inherits all elements of claims 26 and 34. Accordingly, claim 37 is allowable over *Albert* at least for the reasons discussed above with claims 26 and 34. Additionally, claim 37 further recites:

The server computer as described in Claim 34, wherein said method comprises the further steps of:

monitoring TCP/IP control traffic for said communication session at said BTCP module;

understanding when said communication session is closed at said server computer; and

sending a termination message to said first server computer over said control channel.

Albert fails to teach all of the further elements of claim 37. For instance, *Albert* fails to teach at least the recited BTCP module and control channel, and thus fails to teach the above steps that involve such BTCP module and control channel. Accordingly, claim 37 is not anticipated under 35 U.S.C. § 102 by *Albert*. As such, Appellant respectfully requests that the rejection of claim 37 be overturned.

VIII. CLAIMS

A copy of the claims involved in the present appeal is attached hereto as Appendix A.

IX. EVIDENCE

As noted in Appendix B hereto, no evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS

The related appeal identified in Section II is listed in Appendix C attached hereto. No decision has been received for the related appeal, and thus no copy of a decision is provided.

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV 568257480US in an envelope addressed to: M/S Appeal Brief - Patents, Commissioner for Patents, Alexandria, VA 22313.

Date of Deposit: January 5, 2006

Typed Name: Gail L. Miller

Signature: Gail L. Miller

Respectfully submitted,

By: _____

Jody C. Bishop
Attorney/Agent for Applicant(s)
Reg. No. 44,034
Date: September 19, 2005
Telephone No. (214) 855-8007

APPENDIX A

Claims Involved in the Appeal of Application Serial No. 09/880,631

1. In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;

b) handing off said communication session to a selected server computer from said first server computer over a persistent control channel using TCP handoff modules that are dynamically loadable within TCP/IP stacks in operating systems located at both said first server computer and said selected server computer, that implement a TCP handoff protocol that works within kernel levels of an existing TCP/IP protocol; and

c) migrating a first TCP state of said first server computer to said selected server computer, and a second TCP state of said selected server computer to said first server computer over said control channel.

2. The method as described in Claim 1, wherein said step a) comprises the steps of:

receiving a SYN packet from said client at a first BTCP module located at said first server computer;

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer;

receiving a first SYN/ACK packet from said first TCP module;

parsing said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with said TCP/IP communication session;

sending said SYN/ACK packet to said client;

receiving an ACK packet from said client at said first BTCP module;

sending said ACK packet to said first TCP module;

receiving a web request packet associated with said TCP/IP communication session at

said first BTCP module at said first server computer;

storing said SYN, ACK and said web request packet at said first server computer.

3. The method as described in Claim 2, wherein said step b) comprises the steps of:

examining content of said web request packet;

determining which of said plurality of server computers, a selected server computer, can best process said WEB request packet, based on said content;

sending a handoff request from said first BTCP module to a second BTCP module at said selected server computer over said control channel, if said selected server computer is not said first server computer;

including said SYN packet and said ACK packet in said handoff request packet;

changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module;

sending said SYN packet to said second TCP module;

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session;

changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module;

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;

sending said ACK packet that is updated to said second TCP module; and

sending a handoff acknowledgment message to said first BTCP module.

4. The method as described in Claim 3, wherein step c) comprises the steps of:

monitoring traffic associated with establishing said TCP/IP communication session in step a), at said first BTCP module, to parse a first initial TCP state of said first server computer, said first initial TCP state associated with said TCP/IP communication session; and

migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request packet, said first initial TCP state including a first sequence number, such that said second BTCP module can

calculate said first TCP state for said first server computer in said TCP/IP communication session.

5. The method as described in Claim 3, wherein step c) comprises the steps of:

monitoring traffic associated with handing off said TCP/IP communication session at said second BTCP module, to parse a second initial TCP state of said selected server computer, said second initial TCP state associated with said TCP/IP communication session; and

migrating said second initial TCP state of said selected server computer to said first BTCP module by including said second initial TCP state in said handoff acknowledgment packet, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said selected server computer in said TCP/IP communication session.

6. The method as described in Claim 2, comprising the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session; and

holding said connection indication message at said first UTCP module.

7. The method as described in Claim 6, wherein said method comprises the further steps of:

sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module;

discarding said connection indication message at said first UTCP module;

receiving incoming data packets from said client at said first BTCP module;

changing said destination addresses of said incoming data packets to said second IP address;

updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and

forwarding said data packets to said selected server computer.

8. The method as described in Claim 6, comprising the further steps of:

sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer;

sending incoming data packets, including said web request packet, from said client, received at said first BTCP module, upstream.

9. The method as described in Claim 1, comprising the further step of:

intercepting outgoing response packets from said selected server computer at a second bottom TCP (BTCP) module located at said selected server computer;

changing source addresses of said response packets to a first IP address of said first server computer;

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said response packets to said client.

10. The method as described in Claim 1, comprising the further steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module;

understanding when said communication session is closed at said second server computer;

sending a termination message to said first server computer over said control channel;

terminating said TCP/IP communication session at said first server computer by terminating a forwarding mode at said first BTCP module; and

freeing data resources associated with said communication session at said first server computer.

11. In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a TCP/IP communication session between a client computer and a first server computer, said first server computer part of a plurality of server computers forming a web cluster containing information, said communication session established for the transfer of data contained within said information;

b) monitoring traffic associated with establishing said TCP/IP communication

session to understand a first initial TCP state of said first server computer associated with said TCP/IP communication session, at a first bottom-TCP (BTCP) module at said first server computer;

c) receiving a web request associated with said TCP/IP communication session at said first BTCP module at said first server computer;

d) examining content of said web request;

e) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;

f) handing off said communication session to said selected server computer from said first server computer over a persistent control channel, if said selected server computer is not said first server computer;

g) monitoring traffic associated with handing off said TCP/IP communication session to understand a second initial TCP state of said selected server computer associated with said TCP/IP communication session, at a second BTCP module at said selected server computer;

h) migrating said first initial TCP state to said selected server computer over said control channel, such that said second BTCP module can calculate a first TCP state for said first server computer in said TCP/IP communication session;

i) sending a second initial TCP state of said selected server computer to said first BTCP module, such that said first BTCP module can calculate a second TCP state for said selected server computer in said TCP/IP communication session;

j) forwarding data packets received at said first BTCP module from said client to said selected server computer, by changing said data packets to reflect said second TCP state and a second IP address of said selected server computer;

k) sending response packets from said selected server computer directly to said client computer by changing said response packets to reflect said first TCP state and a first IP address of said first server computer; and

l) terminating said TCP/IP communication session at said first server computer when said TCP/IP communication session is closed.

12. The method as described in Claim 11, wherein said step a) comprises the steps of:

receiving a packet from said client at said first BTCP module;

sending said SYN packet upstream to a first TCP module located above said first BTCP module in a first operating system of said first server computer;

receiving a first SYN/ACK packet from said first TCP module;

parsing said first initial TCP state from said first SYN/ACK packet, including a first initial sequence number for said first TCP module associated with, said TCP/IP communication session;

sending said SYN/ACK packet to said client;

receiving an ACK packet from said client at said first BTCP module;

sending said ACK packet to said first TCP module;

storing said SYN, ACK and said web request at said first server computer.

13. The method as described in Claim 11, wherein said step e) comprises the steps of:

sending a handoff request packet from said first BTCP module to said second BTCP module over said control channel;

including said SYN packet and said ACK packet in said handoff request packet;

changing a first destination IP address of said SYN packet to a second IP address of said selected server computer, at said second BTCP module;

sending said SYN packet to said second TCP module;

receiving a second SYN/ACK packet at said second BTCP module;

parsing said second initial TCP state from said second SYN/ACK packet, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session;

changing a second destination IP address of said ACK packet to said second IP address, at said second BTCP module;

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;

sending said ACK packet that is updated to said second TCP module; and

sending a handoff acknowledgment message to said first BTCP module.

14. The method as described in Claim 13, wherein said ACK packet includes said first initial TCP state of said first server computer as provided for in step f).

15. The method as described in Claim 13, wherein said handoff acknowledgment includes said second initial TCP state of said second server computer, including a second initial sequence number, for said second TCP module, that is associated with said TCP/IP communication session as provided for in step i).

16. The method as described in Claim 13, comprising the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session; and

holding said connection indication message at said first UTCP module.

17. The method as described in Claim 16, wherein step h) comprises the further steps of:

sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module;

discarding said connection indication message at said first UTCP module;

receiving incoming data packets from said client at said first BTCP module;

changing said destination addresses of said incoming data packets to said second IP address;

updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and

forwarding said data packets to said selected server computer.

18. The method as described in Claim 11, wherein step k) comprises the steps of:

intercepting outgoing response packets from said selected server computer at said second BTCP module;

changing source addresses of said response packets to said first IP address;

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said updated response packets to said client.

19. The method as described in Claim 11, wherein step 1) comprises the steps of:

monitoring TCP/IP control traffic for said communication session at said second BTCP module;

understanding when said communication session is closed at said second server computer;

sending a termination message to said first server computer over said control channel;

terminating a forwarding mode at said first BTCP module; and

freeing data resources associated with said communication session at said first server computer.

20. The method as described in Claim 16, comprising the further steps of:

sending notification from said first BTCP module to said first UTCP module to release said connection indication message, if said selected server computer is said first server computer; and

sending incoming data packets, including said web request, from said client, received at said first BTCP module, upstream.

21. The method as described in Claim 11, wherein each of said plurality of server computers is constructed similarly including BTCP modules located downstream from TCP modules, and UTCP modules located upstream from TCP modules.

22. The method as described in Claim 12, comprising the further step of storing said web request, said SYN packet, said ACK packet, and said web request at said first server computer.

23. The method as described in Claim 22, wherein said control channel allows for communication between all UTCP modules.

24. The method as described in Claim 11, wherein said plurality of server computers is coupled together over a wide area network in said communication network.

25. The method as described in Claim 11, wherein said information is partitioned/partially replicated throughout each of said plurality of server computers.

26. A server computer comprising:

an upper TCP (UTCP) module located above a TCP module in an operating system of said server computer;

a bottom TCP (BTCP) module located below said TCP module, said UTCP, TCP, and BTCP modules implementing a method of handing off a communication session between a first node and second node in a cluster network that works within the kernel level of an existing TCP/IP protocol, by migrating TCP states associated with said first and second nodes.

27. The server computer as described in Claim 26, wherein said method comprises the steps of:

a) establishing a TCP/IP communication session between a client computer and said server computer, said first node, said server computer part of a plurality of server computers forming said cluster network containing information, said communication session established for the transfer of data contained within said information;

b) receiving a web request associated with said TCP/IP communication session at a first BTCP module at said server computer;

c) examining content of said web request;

d) determining which of said plurality of server computers, a selected server computer, can best process said web request, based on said content;

e) handing off said communication session to said selected server computer from said server computer over a persistent control channel, if said selected server computer is not said server computer; and

f) migrating a first TCP state of said server computer to said selected server computer, and sending a second TCP state of said selected server computer to said server computer over said control channel.

28. The server computer as described in Claim 27, wherein step a) of said method comprises the steps of:

receiving a SYN packet from said client at said BTCP module;

sending said SYN packet upstream to said TCP module;

receiving a first SYN/ACK packet from said TCP module;

parsing a first initial TCP state from said first SYN/ACK packet, including a first

initial sequence number for said TCP module associated with said TCP/IP communication session;

sending said SYN/ACK packet to said client;

receiving an ACK packet from said client at said BTCP module;

sending said ACK packet to said TCP module;

storing said SYN, ACK at said server computer.

29. The server computer as described in Claim 28, wherein said method comprises the steps of:

sending a handoff request packet from said BTCP module to a second BTCP module over said control channel, said second BTCP module located below a second TCP module in a second operating system at said selected server computer;

including said SYN packet and said ACK packet in said handoff request;

receiving a handoff acknowledgment message at said BTCP module from said second BTCP module.

30. The server computer as described in Claim 29, wherein said step f) of said method comprises the steps of:

monitoring traffic associated with establishing said TCP/IP communication session in step a), at said BTCP module, to parse a first initial TCP state of said server computer, said first initial TCP state associated with said TCP/IP communication session; and

migrating said first initial TCP state to said second BTCP module over said control channel by including said first initial TCP state in said handoff request, said first initial TCP state including a first sequence number, such that said second BTCP module can calculate said first TCP state for said server computer in said TCP/IP communication session.

31. The server computer as described in Claim 29, wherein said method comprises the further steps of:

intercepting a connection indication message sent from said first TCP module to an application layer above said first TCP module at a first upper-TCP (UTCP) module, said connection indication message sent by said first TCP module upon establishing said communication session; and

holding said connection indication message at said first UTCP module.

32. The computer system as described in Claim 31, wherein said method comprises the further steps of:

sending a reset packet from said first BTCP module upon receiving said handoff acknowledgment packet to said first TCP module;

discarding said connection indication message at said first UTCP module;

receiving incoming data packets from said client at said first BTCP module;

changing said destination addresses of said incoming data packets to said second IP address;

updating sequence numbers and TCP checksum in said data packets to reflect said second TCP state of said selected server computer; and

forwarding said data packets to said selected server computer.

33. The server computer as described in Claim 31, said method comprising the further steps of:

sending notification from said BTCP module to said UTCP module to release said connection indication message, if said selected server computer is said server computer;

sending incoming data packets, including said web request, from said client, received at said first BTCP module, upstream.

34. The server computer as described in Claim 26, said method comprising the further steps of:

receiving a handoff request from a first BTCP module located at a first server computer within said cluster network over a persistent control channel, said first server computer having established a communication session with a client computer, said communication session established for the transfer of data contained within said server computer, said handoff request including a SYN packet and an ACK packet, said SYN and ACK packet used for establishing said communication session between said client and said first server computer, said ACK packet including a first initial TCP state of said first server computer in said communication session, including a first initial TCP sequence number;

changing a first destination IP address of said SYN packet to a second IP address of said server computer, at said BTCP module;

sending said SYN packet to said TCP module;

receiving a SYN/ACK packet at said second BTCP module;

parsing a second initial TCP state from second SYN/ACK packet, including a second initial sequence number, for said TCP module, said second initial TCP state associated with a second TCP state for said server computer in said TCP/IP communication session;

changing a second destination IP address of said ACK packet to said second IP address, at said BTCP module;

updating said ACK packet to reflect said second TCP state of said selected server computer in said communication session;

sending said ACK packet that is updated to said TCP module; and

sending a handoff acknowledgment message to said first BTCP module over said control channel.

35. The server computer as described in Claim 34, wherein said method comprises the further steps of:

monitoring traffic associated with handing off said TCP/IP communication session to said server computer, at said BTCP module, to parse said second initial TCP state of said server computer, said second initial TCP state associated with said TCP/IP communication session; and

sending said second initial TCP state of said server computer to said first BTCP module by including said second initial TCP state in said handoff acknowledgment, said second initial TCP state including a second initial sequence number, such that said first BTCP module can calculate said second TCP state for said server computer in said TCP/IP communication session.

36. The server computer as described in Claim 34, wherein said method comprises the further steps of:

intercepting outgoing response packets from said server computer at said second BTCP module;

changing source addresses of said response packets to said first IP address;

updating sequence numbers and TCP checksum in said response packets to reflect said first TCP state of said first server computer; and

sending said response packets to said client.

37. The server computer as described in Claim 34, wherein said method comprises the further steps of:

monitoring TCP/IP control traffic for said communication session at said BTCP module;

understanding when said communication session is closed at said server computer; and

sending a termination message to said first server computer over said control channel.

APPENDIX B

Evidence

None.

APPENDIX C

Related Proceedings

A currently pending appeal before the Board of co-pending U.S. Patent Application Serial No. 09/880,632, and particularly the Board's interpretation of *Albert*, may affect, be affected by, or have a bearing on the Board's decision in this appeal.

Application No. (if known): 09/880,632 Attorney Docket No. 10010812-1

Certificate of Express Mailing Under 37 CFR 1.10

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Airbill No. EV 568257480US in an envelope addressed to:

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

on January 5, 2006
Date

Signature

Gail Miller
Typed or printed name of person signing Certificate

Registration Number, if applicable

(214) 855-8379
Telephone Number

Note: Each paper must have its own certificate of mailing, or this certificate must identify each submitted paper.

2 Return Postcards
Transmittal – 1 page
Corrected Appeal Brief – 53 pages
Certificate of Mail – 1 page